Discrete Sampling using Semigradient-based Product Mixtures

Alkis Gotovos
ETH Zurich
alkisg@inf.ethz.ch

Hamed Hassani
University of Pennsylvania
hassani@seas.upenn.edu

Andreas Krause
ETH Zurich
krausea@ethz.ch

Stefanie Jegelka
MIT
stefje@mit.edu

Abstract

We consider the problem of inference in discrete probabilistic models, that is, distributions over subsets of a finite ground set. These encompass a range of well-known models in machine learning, such as determinantal point processes and Ising models. Locally-moving Markov chain Monte Carlo algorithms, such as the Gibbs sampler, are commonly used for inference in such models, but their convergence is, at times, prohibitively slow. This is often caused by state-space bottlenecks that greatly hinder the movement of such samplers. We propose a novel sampling strategy that uses a specific mixture of product distributions to propose global moves and, thus, accelerate convergence. Furthermore, we show how to construct such a mixture using semigradient information. We illustrate the effectiveness of combining our sampler with existing ones, both theoretically on an example model, as well as practically on three models learned from real-world data sets.

1 INTRODUCTION

Discrete probabilistic models have played a fundamental role in machine learning. Examples range from classic graphical models, such as Ising and Potts models (Koller and Friedman, 2009), which have long been used in computer vision applications (Boykov et al., 2001), to determinantal point processes (Kulesza and Taskar, 2012) used in video summarization (Gong et al., 2014), and facility location diversity models used for product recommentation (Tschiatschek et al., 2016). Recently, there has been increased interest in general distributions over subsets of a finite ground set V; that is, given a set function $F : 2^V \to \mathbb{R}$, distributions of the form $\pi(S) \propto \exp(F(S))$, for all $S \subseteq V$. These can be equivalently seen as distributions over binary random vectors, if S is replaced by the indicator function of the corresponding vector. All the aforementioned examples can be expressed in this form for a suitable choice of F.

While exact inference in such models is known to be intractable in general (Jerrum and Sinclair, 1993), there has been recent work on analyzing approximate inference techniques, such as variational methods (Djolonga and Krause, 2014; Djolonga et al., 2016b), and Markov chain Monte Carlo (MCMC) sampling (Gotovos et al., 2015; Rebeschini and Karbasi, 2015). The sampling analyses, in particular, focus on the Gibbs sampler, and derive sufficient conditions under which it mixes—converges toward the target distribution—sufficiently fast.

Unfortunately, oftentimes in practice these conditions do not hold and the Gibbs sampler mixes prohibitively slowly. A fundamental reason for this slow mixing behavior is the existence of bottlenecks in the state space of the Markov chain. Conceptually, one can think about the state-space graph containing several isolated components that are poorly connected to each other, thus making it hard for the Gibbs sampler to move between them.

In this work, we propose a novel sampling strategy that allows for global moves in the state space, thereby avoiding bottlenecks, and, thus, accelerating mixing. Our sampler is based on using a proposal distribution that approximates the target π by a mixture of product distributions. We further propose an algorithm for constructing such a mixture using discrete semigradient information of the associated function F. This idea makes a step towards bridging optimization and sampling, a theme that has been successful in continuous spaces. Our sampler is readily combined with other existing samplers, and we show provable theoretical, as well as empirical examples of speedups.

Contributions. The main contributions of this paper are as follows.

- We propose the M^3 sampler, which makes global moves according to a specific mixture of product distributions.
- We theoretically analyze mixing times on an illustrative family of Ising models, and prove that adding the M^3 sampler results in an exponential improvement over the Gibbs sampler.
- We demonstrate the effectiveness of combining the M^3 and Gibbs samplers in practice on three models learned from real-world data.

Related work. Recent work on analyzing the mixing time of MCMC samplers for discrete probabilistic models includes deriving general conditions on F to achieve fast mixing (Gotovos et al., 2015; Rebeschini and Karbasi, 2015; Li et al., 2016), as well as looking at specific subclasses, such as strongly Rayleigh distributions (Li et al., 2016; Anari et al., 2016).

There has also been work on mapping discrete inference to continuous domains (Zhang et al., 2012; Pakman and Paninski, 2013; Dinh et al., 2017; Nishimura et al., 2018) to enable the use of well-established continuous samplers, such as Hamiltonian Monte Carlo (Neal, 2012; Betancourt, 2017). It is worth pointing out that, while these methods usually outperform simple Gibbs or Metropolis samplers, they still tend to suffer from considerable slowdowns in multimodal distributions (Neal, 2012). Our work is orthogonal to these methods, in the sense that our proposed sampler can be combined with any of the existing ones to provide a principled way for performing global moves that can lead to improved mixing.

Both darting Monte Carlo (Sminchisescu and Welling, 2007; Ahn et al., 2013) and variational MCMC (de Freitas et al., 2001) share the high-level concept of combining two chains, one making global moves between high-probability regions, and another making local moves around those regions. However, their proposed global samplers for continuous spaces are generally not applicable to the class of discrete distributions we consider.

There are several well-known results on mixing of the Gibbs sampler for the Ising model on different graph structures (Jerrum and Sinclair, 1993; Berger et al., 2005; Levin et al., 2008a;b). Other (non-MCMC) approaches to discrete sampling include Perturb-and-MAP (Papandreou and Yuille, 2011; Hazan et al., 2013), and random projections (Zhu and Ermon, 2015). Semigradients of submodular set functions have recently been exploited for optimization (Iyer et al., 2013; Jegelka and Bilmes, 2011) and variational inference (Djolonga et al., 2016a), but, to our knowledge, no prior work has used them for sampling.

2 BACKGROUND

We consider set functions $F : 2^V \to \mathbb{R}$, where V is a finite ground set of size n that can be assumed to be $V = \{1, \dots, n\}$ without loss of generality. In this paper, we focus on distributions over $\Omega := 2^V$ of the form

$$\pi(S) = \frac{1}{Z} \exp\left(F(S)\right), \tag{1}$$

for all $S \in \Omega$. The partition function $Z := \sum_{S \in \Omega} \exp(F(S))$ serves as the normalizer of the distribution. Alternatively, we can describe distributions of the above form via binary vectors $X \in \{0,1\}^n$. If we define $V(X) := \{v \in V \mid X_v = 1\}$, then the distribution $p_X(X) \propto \exp(F(V(X)))$ over binary vectors is isomorphic to the distribution (1) over sets.

Perhaps the simplest family of such models are log-modular distributions, which describe a collection of independent binary random variables. Equivalently, they are distributions of the form (1) where F is a modular function, that is, a function of the form $F(S) = c + \sum_{v \in S} m_v$, where $c, m_v \in \mathbb{R}$, for all $v \in V$. The partition function of a log-modular distribution can be derived in closed form as $Z_m = \exp(c) \prod_{v \in V} (1 + \exp(m_v))$. Consequently, the corresponding log-modular distribution is

$$\pi_m(S) = \frac{\exp\left(\sum_{v \in S} m_v\right)}{\prod_{v \in V} (1 + \exp(m_v))}.$$

Inference and sampling. Performing exact inference in models of the form (1), that is, computing conditional probabilities such as $\pi(A \subseteq S \subseteq B \mid C \subseteq S \subseteq D)$, is known to be in general #P-hard (Jerrum and Sinclair, 1993). As a result, we have to resort to approximate inference algorithms, such as Markov chain Monte Carlo sampling (Levin et al., 2008b), which is the primary focus of this paper. An MCMC algorithm for distribution π simulates a Markov chain in state space Ω in such a way that the sequence of visited states $(X_0, X_1, \dots) \in \Omega^{\mathbb{N}}$ ultimately converges to π.

Gibbs sampler. One of the most commonly used chains is the (single-site) Gibbs sampler, which adds or removes a single element at a time. It first selects uniformly at random an element $v \in V$; subsequently, it adds or removes v to the current state X_t according to the probability of the resulting state. We denote by $P : \Omega \times$

$\Omega \to \mathbb{R}$ the transition matrix of a Markov chain, that is, for all $S, R \in \Omega$, $P(S,R) := \mathbb{P}[X_{t+1} = R \mid X_t = S]$. Then, if we define

$$p_{S \to R} = \frac{\exp(F(R))}{\exp(F(R)) + \exp(F(S))},$$

and denote by $S \sim R$ states that differ by exactly one element (i.e., $\big||R| - |S|\big| = 1$), the transition matrix P^{G} of the Gibbs sampler is

$$P^{\mathrm{G}}(S,R) = \begin{cases} \frac{1}{n} p_{S \to R}\,, & \text{if } R \sim S \\ 1 - \sum_{T \sim S} \frac{1}{n} p_{S \to T}\,, & \text{if } R = S \\ 0\,, & \text{otherwise} \end{cases}.$$

Mixing. The efficiency of a Markov chain in approximating its target distribution depends largely on the speed of convergence of the chain, which is quantified by the chain's mixing time. Most commonly, distance from stationarity is measured by the maximum total variation distance, over all starting states, between X_t and the target distribution π, that is, $d(t) := \max_{X_0 \in \Omega} d_{\mathrm{TV}}\left(P^t(X_0, \cdot), \pi\right)$. Then, the mixing time denotes the minimum number of iterations required to get ϵ-close to stationarity, $t_{\mathrm{mix}}(\epsilon) := \min\{t \mid d(t) \le \epsilon\}$.

A common way to obtain an upper bound on the mixing time of a chain is by lower bounding its spectral gap, defined as $\gamma := 1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the transition matrix P. The following well-known theorem connects the spectral gap to mixing time.

Theorem 1 (cf. Theorems 12.3, 12.4 in (Levin et al., 2008b))**.** *Let P be the transition matrix of a lazy, irreducible, and reversible Markov chain, and let γ be its spectral gap, and $\pi_{min} := \min_{S \in \Omega} \pi(S)$. Then,*

$$\left(\frac{1}{\gamma} - 1\right) \log\left(\frac{1}{2\epsilon}\right) \le t_{\mathrm{mix}}(\epsilon) \le \frac{1}{\gamma} \log\left(\frac{1}{\epsilon \pi_{min}}\right).$$

3 THE MIXTURE CHAIN

Despite the simplicity and computational efficiency of the Gibbs sampler, the fact that it is constrained to performing local moves makes it susceptible to state-space bottlenecks, which hinder the movement of the chain around the state space. Intuitively, the state space may contain several high-probability regions arranged in such a way that moving from one to another using only single-element additions and deletions requires passing through states of very low probability. As a result, the Gibbs sampler may mix extremely slowly on the whole state space, despite the fact that it can move sufficiently fast within each of the high-probability regions.

To alleviate this shortcoming, it is natural to ask whether it is possible to bypass such bottlenecks by using a chain that performs larger moves. In this paper, we introduce a novel approach that uses a Metropolis chain based on a specific mixture of log-modular distributions, which we call the M^3 chain, to perform global moves in state space. Concretely, we define a proposal distribution

$$\begin{aligned} q(S,R) = q(R) &= \frac{1}{Z_q} \sum_{i=1}^{r} \exp\left(F_i(R)\right) \\ &= \frac{1}{Z_q} \sum_{i=1}^{r} w_i \exp\left(m_i(R)\right), \end{aligned} \tag{2}$$

where each $F_i(R) = c_i + \sum_{v \in R} m_{iv}$ is a modular function, while each $m_i(R) = \sum_{v \in R} m_{iv}$ is a normalized modular function ($m_i(\emptyset) = 0$), and $w_i = \exp(c_i) > 0$. If we denote by Z_i the normalizer of m_i, then the normalizer of the mixture can be written in closed form as

$$\begin{aligned} Z_q = \sum_{R \in \Omega} q(R) &= \sum_{R \in \Omega} \sum_{i=1}^{r} w_i \exp\left(m_i(R)\right) \\ &= \sum_{i=1}^{r} w_i \sum_{R \in \Omega} \exp\left(m_i(R)\right) \\ &= \sum_{i=1}^{r} w_i Z_i. \end{aligned}$$

We define the M^3 chain as a Metropolis chain (Levin et al., 2008b) using q as a proposal distribution; its transition matrix $P^{\mathrm{M}} : \Omega \times \Omega \to \mathbb{R}$ is given by

$$P^{\mathrm{M}}(S,R) = \begin{cases} q(R) p_a(S,R)\,, & \text{if } R \neq S \\ 1 - \sum_{T \neq S} q(T) p_a(S,T)\,, & \text{otherwise} \end{cases},$$

where

$$p_a(S,R) := \min\left\{1, \frac{\pi(R) q(S)}{\pi(S) q(R)}\right\}.$$

Note that, contrary to usual practice, the proposal q only depends on the proposed state, but not on the current state of the chain. As a result, the chain is not constrained to local moves, but rather can potentially jump to any part of the state space. In practice, M^3 sampling proceeds in two steps: first, a candidate set R is sampled according to q; then, the move to R is accepted with probability p_a. Sampling from q can be done in $\mathcal{O}(n)$ time—first, sample a log-modular component, then sample a set from that component. Computing p_a requires $\mathcal{O}(r)$ time for the sum in (2), and it can be straightforwardly improved by parallelizing this computation. All in all, the total time for one step of M^3 is $\mathcal{O}(n + r)$.

As is always the case with Metropolis chains, the mixing time of the M^3 sampler will depend on how well the proposal q approximates the target distribution π. The following observation shows that, in theory, we can approximate any distribution of the form (1) by a mixture of the form (2).

Proposition 1. *For any π on Ω as in (1), and any $\epsilon > 0$, there are positive constants $w_i = w_i(\epsilon) > 0$, and normalized modular functions $m_i = m_i(\epsilon)$, such that, if we define $q(S) := \sum_{i=1}^{r} w_i \exp(m_i(S))$, for all $S \in \Omega$, then $d_{TV}(\pi, q) \leq \epsilon$.*

Conceptually, the proof relies on having one log-modular term per set in Ω.[1] Therefore, while the above result shows that mixtures of log-modulars are expressive enough, the constructed mixture of exponential size in n is not useful for practical purposes. On the other hand, it is not necessary for us to have q be an accurate approximation of π everywhere, as long as the corresponding M^3 chain is able to bypass state-space bottlenecks. With this in mind, we suggest combining the M^3 and Gibbs chains, so that each of them serve complementary purposes in the final chain; the role of M^3 is to make global moves and avoid bottlenecks, while the role of Gibbs is to move fast within well-connected regions of the state space. To make this happen, we define the transition matrix $P^{\mathrm{C}} : \Omega \times \Omega \to \mathbb{R}$ of the combined chain as

$$P^{\mathrm{C}}(S, R) = \alpha P^{\mathrm{G}}(S, R) + (1 - \alpha) P^{\mathrm{M}}(S, R), \quad (3)$$

where $0 < \alpha < 1$. It is easy to see that P^{C} is reversible, and has stationary distribution π.

We next illustrate how combining the two chains works on a simple example, where a mixture of only a few log-modular distributions can dramatically improve mixing compared to running the vanilla Gibbs chain. Then we propose an algorithm for automatically creating such a mixture.

3.1 EXAMPLE: ISING MODEL ON THE COMPLETE GRAPH

We consider the Ising model on a finite complete graph (Levin et al., 2008a), also known as the Curie-Weiss model in statistical physics, which can be written in the form of (1) as follows:

$$\pi_\beta(S) = \frac{1}{Z(\beta)} \exp\left(-\frac{2\beta}{n}|S|(n - |S|)\right). \quad (\textsc{Ising}_\beta)$$

[1] Detailed proofs of all our results can be found in the appendix.

In particular, we focus on the case where $\beta = \ln(n)$, that is,

$$\pi(S) = \frac{1}{Z} \exp\left(-\frac{2\ln(n)}{n}|S|(n - |S|)\right). \quad (\textsc{Ising})$$

In this case, if we define $d_n := 2\ln(n)/n$, then $F(S) = -d_n|S|(n - |S|)$.

The Gibbs sampler is known to experience poor mixing in this model; the following is an immediate corollary of Theorem 15.3 in (Levin et al., 2008b).

Corollary 1 (cf. Theorem 15.3 in (Levin et al., 2008b))**.** *For $n \geq 3$, the Gibbs sampler on* ISING *has spectral gap $\gamma^{\mathrm{G}} = \mathcal{O}(e^{-cn})$, where $c > 0$ is a constant.*

From Theorem 1 it follows that the mixing time of Gibbs is $t_{\mathrm{mix}}(\epsilon) = \Omega\left((e^{cn} - 1)\log(1/(2\epsilon))\right)$. Yet, it has been shown that the only reason for this is a single bottleneck in the state space (Levin et al., 2008a). To make this statement more formal, let us define a decomposition of Ω into two disjoint sets, $\Omega_0 := \{S \in \Omega \mid |S| < n/2\}$, and $\Omega_1 := \{S \in \Omega \mid |S| > n/2\}$ (Jerrum et al., 2004). To keep things simple, we will assume for the remainder of this section that n is odd; the analysis when n is even follows from the same arguments with only a minor technical adjustment. Our goal is to separately examine two characteristics of the sampler: (i) its movement between the two sets Ω_0, Ω_1, and (ii) its movement when restricted to stay within each of these sets.

For analyzing the "between-sets" behavior, we define the projection $\bar{\pi} : \{0, 1\} \to \mathbb{R}$ of π as

$$\bar{\pi}(i) := \sum_{S \in \Omega_i} \pi(S),$$

and, for any reversible chain P, we define its projection chain $\bar{P} : \{0, 1\} \times \{0, 1\} \to \mathbb{R}$ as

$$\bar{P}(i, j) := \frac{1}{\bar{\pi}(i)} \sum_{S \in \Omega_i, R \in \Omega_j} \pi(S) P(S, R).$$

It is easy to see that $\bar{P}$ is also reversible and has stationary distribution $\bar{\pi}$. For analyzing the "within-set" behavior, we define the restrictions $\pi_i : \Omega_i \to \mathbb{R}$ of π as

$$\pi_i(S) := \frac{\pi_i(S)}{\bar{\pi}(i)},$$

and the two restriction chains $P_i : \Omega_i \times \Omega_i \to \mathbb{R}$ of P as

$$P_i(S, R) := \begin{cases} P(S, R), & \text{if } S \neq R \\ 1 - \sum_{T \in \Omega_i : T \neq S} P(S, T), & \text{otherwise} \end{cases}.$$

Again, it is easy to see that each of the P_i is also reversible and has stationary distribution π_i.

Coming back to the Gibbs sampler, if we could show that it mixes fast within each of Ω_0 and Ω_1, then we could deduce that the only reason for the slow mixing on Ω is the bottleneck between these two sets. Indeed, the following corollary of a theorem by Ding et al. (2009) shows exactly that.

Corollary 2 (cf. Theorem 2 in (Ding et al., 2009))**.** *For all $n \geq 3$, the restriction chains of the Gibbs sampler P_i^{G}, $i = 0, 1$, on* ISING *have spectral gap $\gamma_i^{\mathrm{G}} = \Theta\big(\frac{2\ln(n)-1}{n}\big)$.*

To improve mixing we want to create an M^3 chain that is able to bypass the aforementioned bottleneck. For this purpose, we use a mixture of two log-modular distributions, the first of which puts most of its mass on Ω_0, and the second on Ω_1. We define the mixture of the form (4) by

$$m_1(S) = \sum_{v \in S} -d_n(n-1) = -d_n(n-1)|S|,$$
$$m_2(S) = \sum_{v \in S} d_n(n-1) = d_n(n-1)|S|.$$

We also use $w_1 = 1/Z_1$ and $w_2 = 1/Z_2$, where Z_1 and Z_2 are the normalizers of m_1 and m_2 respectively. It follows that $Z_q = 1/2$, and, furthermore, the mixture q is symmetric, that is, $q(S) = q(V \setminus S)$. Since the proposal q is symmetric and state independent, we would expect the M^3 chain to jump between Ω_0 and Ω_1 without being hindered by the bottleneck described previously. We verify this intuition by proving the following lemma.

Lemma 1. *For all $n \geq 10$, the projection chain $\bar{P}^{\mathrm{M}}$ of the M^3 sampler on* ISING *has spectral gap $\bar{\gamma}^{\mathrm{M}} = \Omega(1)$.*

Putting everything together we show the following result about the combined chain P^{C}.

Theorem 2. *For all $n \geq 10$, the combined chain P^{C} on* ISING *has spectral gap*

$$\gamma^{\mathrm{C}} = \Omega\left(\frac{2\ln(n)-1}{2n}\right).$$

The proof consists of two steps. In the first step we make a comparison argument (Diaconis and Saloff-Coste, 1993; Levin et al., 2008b) to show that the spectral gaps of the projection and restriction chains of the combined sampler are smaller by at most a constant factor in α compared to those of Gibbs and M^3. In particular, we use the M^3 bound (Lemma 1) for the projection chain, and the Gibbs bound (Theorem 2) for the restriction chains. The second step, then, combines the projection and restriction bounds to establish a bound on the spectral gap of the combined chain. To accomplish this we use a result by Jerrum et al. (2004), which, roughly speaking, states that the spectral gap of the whole chain cannot be much smaller than the smallest of the projection and restriction spectral gaps.

Algorithm 1 Iterative semigradient-based mixture construction

Input: Set function F, mixture size r
for $i = 1$ **to** r **do**
  $\sigma \leftarrow \textsc{Greedy}(F, \{m_1, \ldots, m_{i-1}\})$
  $m_i \leftarrow \textsc{SemiGradient}(F, \sigma)$
return $\{m_1, \ldots, m_r\}$

Finally, using Theorem 1, and noting that, in this case, $\pi_{\min} = \mathcal{O}(e^{-n})$ (cf. proof of Lemma 1), we get a mixing time of $t_{\mathrm{mix}}(\epsilon) = \mathcal{O}(n^2 \log(1/\epsilon))$ for the combined chain. This shows that the addition of the M^3 sampler results in an exponential improvement in mixing time over the Gibbs sampler by itself.

4 CONSTRUCTING THE MIXTURE

Having seen the positive effect of the M^3 sampler, we now turn to the issue of how to choose the proposal q. While a manual construction like the one we just presented for the Ising model may be feasible in some cases, it is often more practical to have an automated way of obtaining the mixture.

Let us assume, as is usually the case, that we have access to a function oracle for F, and we want to create a mixture of size r. Ideally, we would like to construct a proposal q that is as close to π as possible, that is, minimize an objective such as the following,

$$E_1(q) := \min_q \|\pi - q\| = \min_q \left\| \frac{\exp(F(\cdot))}{Z} - \frac{1}{Z_q} \textstyle\sum_{i=1}^r w_i \exp(m_i(\cdot)) \right\|,$$

where $\|\cdot\|$ could be, for example, total variation distance or the maximum norm. Unfortunately, this problem is hard: both computing the partition function Z, and jointly optimizing over all w_i, m_i are infeasible in practice. To make the problem easier, we could try to get rid of the normalizers and weights w_i, and iteratively minimize over each m_i individually:

$$E_2^{(i)}(m_i) := \min_{m_i} \left\| \exp(F(\cdot)) - \textstyle\sum_{j=1}^{i-1} \exp(m_i(\cdot)) \right\|,$$

for $i \in \{1, \ldots, r\}$. This problem is still hard, since optimizing $\|\exp(F(\cdot))\|$ is by itself infeasible in general.

To arrive at a practical algorithm, we approximate the above objective using the two-step procedure described in Algorithm 1. In the first step, we generate a permutation σ of the ground set V by running

Algorithm 2 Greedy difference maximization

Input: Set function F, modular functions $\{m_1, \dots, m_{i-1}\}$

$D_i(S) \leftarrow F(S) - \log \sum_{j=1}^{i-1} \exp(m_j(S))$, for all $S \in \Omega$
$\sigma \leftarrow (1, \dots, n)$
$A \leftarrow \emptyset$
for $i = 1$ **to** n **do**
  $v^* \leftarrow \operatorname{argmax}_{v \in V} \left(D_i(A \cup \{v\}) - D_i(A)\right)$
  $\sigma_i \leftarrow v^*$
  $A \leftarrow A \cup \{v^*\}$
return σ

the greedy algorithm on function $D_i(S) := F(S) - \log \sum_{j=1}^{i-1} \exp(m_j(S))$, as shown in Algorithm 2. Intuitively, the sets that are formed by elements near the beginning of σ are those on which F and the current mixture disagree by the most. Therefore, in the second step, we would like to add to the mixture a modular function m_i that is a good approximation for F on $\{\sigma_1, \dots, \sigma_k\}$, for a choice of $1 \leq k \leq n$. To accomplish this, we propose using discrete semigradients.

Semigradients are modular functions that provide lower (subgradient) or upper (supergradient) approximations of a set function F (Fujishige, 2005; Iyer et al., 2013). More concretely, given a set $S \in \Omega$, a modular function m is a subgradient of F at S, if, for all $R \in \Omega$, $F(R) \geq F(S) + m(R) - m(S)$. Similarly, m is a supergradient if the inequality is reversed. Although, in general, a function is not guaranteed to have sub- or supergradients at each $S \in \Omega$, it has been shown that this is true when F is submodular or supermodular (Fujishige, 2005; Jegelka and Bilmes, 2011; Iyer and Bilmes, 2012).

Submodularity expresses a notion of diminishing returns; that is, adding an element to a larger set provides less benefit than adding that same element to a smaller set. More formally, F is submodular if, for any $S \subseteq R \subseteq V$, and any $v \in V \setminus R$, it holds that $F(R \cup \{v\}) - F(R) \leq F(S \cup \{v\}) - F(S)$. Supermodularity is defined in a similar way by reversing the sign of this inequality. The resulting models of the form (1) are referred to as log-submodular and log-supermodular respectively. Many commonly used models fall under these categories; Ising and Potts models, including our example in the previous section, are log-supermodular, while determinantal point processes and facility location diversity models are log-submodular.

Coming back to the second step of Algorithm 1, to create a subgradient of F given permutation σ we just need to define a modular function via marginal gains according to the permutation order (Iyer et al., 2013), as shown in Algorithm 3. Moreover, this is a subgradient of F at $\{\sigma_1, \dots, \sigma_k\}$, for all $1 \leq k \leq n$. On the other hand,

Algorithm 3 Subgradient computation

Input: Set function F, permutation σ

$A \leftarrow \emptyset$
$f \leftarrow F(\emptyset)$
for $v = 1$ **to** n **do**
  $m_v \leftarrow F(A \cup \{\sigma_v\}) - F(A)$
  $A \leftarrow A \cup \sigma_v$
return $m(S) := \sum_{v \in S} m_v$, for all $S \in \Omega$

Algorithm 4 Supergradient computation

Input: Set function F, permutation σ

$k \leftarrow$ DRAWUNIFORM(1, n)
for $v = 1$ **to** k **do**
  $m_v \leftarrow F(V) - F(V \setminus \{v\})$
for $v = k + 1$ **to** n **do**
  $m_v \leftarrow F(\{v\})$
return $m(S) := \sum_{v \in S} m_v$, for all $S \in \Omega$

Algorithm 4 creates a supergradient of F at $\{\sigma_1, \dots, \sigma_k\}$ for a randomly chosen k. (This type of supergradient is denoted by $\bar{g}_Y$ by Iyer et al. (2013).) In fact, the modular functions m_1, m_2 that we used in analyzing the Ising model in the previous section were supergradients of F at sets $S_1 = \emptyset$, and $S_2 = V$ respectively.

In practice, we can use Algorithm 1 regardless of whether F is sub- or supermodular. We have, however, noticed that subgradients give better results when F is submodular, and the same goes for supergradients and supermodular functions.

5 EXPERIMENTS

We now evaluate the performance of our proposed sampler on the Ising model we analyzed earlier, as well as the following three models learned from real-world data sets.

WATER. A (log-submodular) facility location model, which was used in a problem of sensor placement in a water distribution network (Krause et al., 2008). The function F is of the form

$$F(S) = \sum_{j=1}^{L} \max_{i \in S} c_{ij}.$$

We randomly subsample the original facility location matrix $C = (c_{ij})$, so that $n = 50$, and $L = 500$.

SENSOR. A (log-submodular) determinantal point process (Kulesza and Taskar, 2012), which was used in a problem of sensor placement for indoor temperature

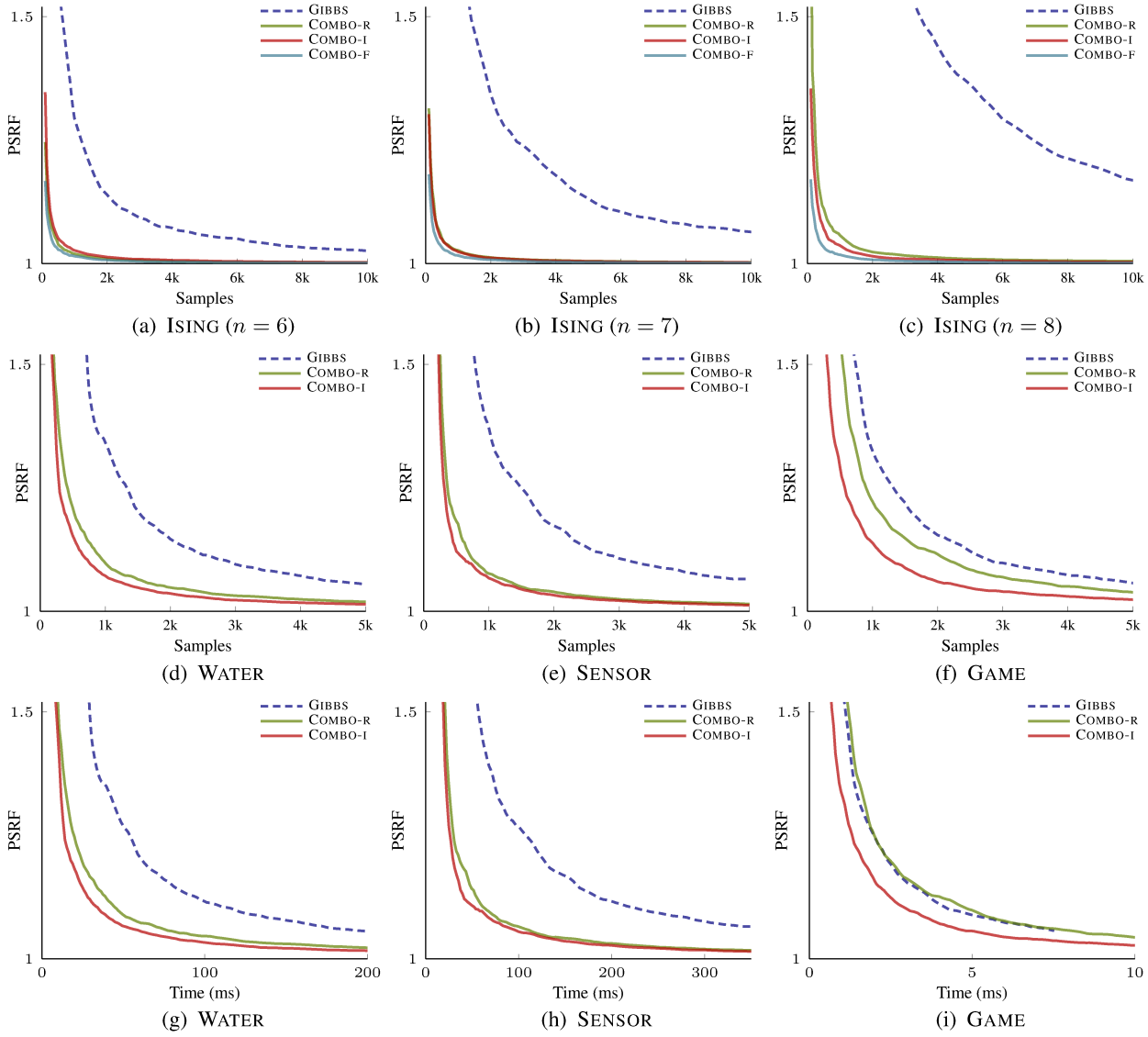


Figure 1: (a)-(c) Ising model results for increasing n. Note how the Gibbs sampler gets worse significantly faster than the combined ones. (d)-(f) Potential scale reduction factor (PSRF) as a function of sampling iterations. (g)-(i) PSRF as a function of wall-clock time in milliseconds. The combined sampler outperforms Gibbs both in terms of samples required, as well as actual runtime.

monitoring (Guestrin et al., 2005). The function F is of the form

$$F(S) = \log |K + \sigma^2 I|,$$

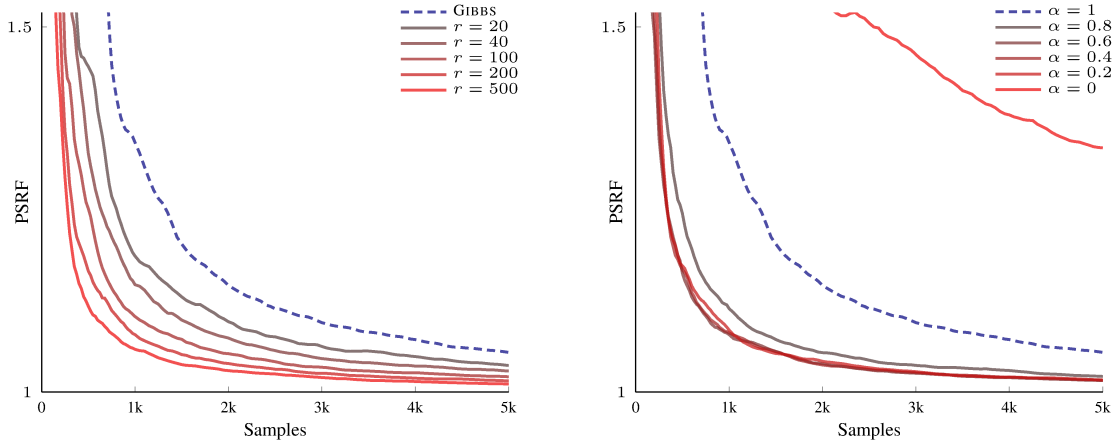
where K is a kernel matrix, and σ is a noise parameter. The size of the ground set is $n = 46$.

GAME. A (log-submodular) facility location diversity model (Tschiatschek et al., 2016), which represents the characters that are chosen by players in the popular online game "Heroes of the Storm". We learned the model from an online data set of approximately $8,000$ teams of 5 characters[2] using noise-contrastive estimation, as described by Tschiatschek et al. (2016). The function F is of the form

$$F(S) = \sum_{v \in S} w_v + \sum_{j=1}^{L} \max_{i \in S} c_{ij},$$

with $n = 48$, and $L = 10$. In practice, we would only be interested in sampling sets of fixed size $\ell = 5$. The Gibbs sampler can be easily modified to sample under a cardinality constraint by using moves that swap an element in

[2] https://www.hotslogs.com



(a) WATER (b) WATER

Figure 2: (a) Increasing the number of mixture components improves performance. (b) The combination of Gibbs and M^3 performs better than either of them does individually.

the current set X_t with an element in $V \setminus X_t$. Extending the M^3 chain to sample from cardinality-constrained models is also straightforward. In fact, the only additional ingredient required is a procedure to sample a set of size ℓ from a log-modular distribution, which can be easily done, as before, in $\mathcal{O}(n)$ time.

In what follows, we compare the performance of the Gibbs sampler (GIBBS) against our proposed combined sampler using a proposal mixture q constructed by Algorithm 1 (COMBO-I). We also compare against a variation where we substitute the greedy procedure in line 2 of Algorithm 1 with picking a permutation σ of the ground set uniformly at random (COMBO-R).

To assess convergence we use the potential scale reduction factor (PSRF) (Brooks et al., 2011) using 20 parallel chains. We compute the PSRF using single-element marginal probabilities averaged over 50 repetitions of each simulation.

In Figures 1a–1c we show the results for the Ising model ($n = 6, 7, 8$) with the additional COMBO-F line denoting the combined sampler with two mixture components described in Section 3.1. The other two combined samplers use mixtures of size $r = 20$. Note that Gibbs mixes dramatically slower than the combined sampler, even for such small n.

In Figures 1d–1f we show the results on the three log-submodular models described before using mixtures of size $r = 200$. It is interesting to see that even random permutations are enough to significantly improve over the performance of Gibbs. Similar observations can be made with respect to computation time, as shown in Figures 1g–1i, which measure wall-clock time on the x-axis.

In Figure 2a we show how mixture size affects performance; as expected, adding more components to the mixture results in a proposal that approximates the target distribution better, and, therefore, mixes faster. Finally, in Figure 2b we see that both Gibbs ($\alpha = 1$) and M^3 ($\alpha = 0$, $r = 200$) perform poorly by themselves, but combining them results in much improved performance. This highlights again the complementary nature of the two chains (local vs. global moves) we discussed earlier.

6 CONCLUSION

We considered the problem of sampling from general discrete probabilistic models, and presented the M^3 sampler that proposes global moves using a mixture of log-modular distributions. We theoretically analyzed the effect of combining our sampler with the Gibbs sampler on a class of Ising models, and proved an exponential improvement in mixing time. We also demonstrated notable improvements when combining the two samplers on three models of practical interest. We believe that our work represents a step towards moving beyond local samplers, and incorporating ideas from optimization, such as semigradients, into probabilistic inference.

Acknowledgements

This work was partially supported by ERC Starting Grant 307036, NSF CAREER award 1553284, and the Simons Institute for the Theory of Computing. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

S. Ahn, Y. Chen, and M. Welling. Distributed and adaptive darting monte carlo through regenerations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

N. Anari, S. O. Gharan, and A. Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *Conference on Learning Theory (COLT)*, 2016.

N. Berger, C. Kenyon, E. Mossel, and Y. Peres. Glauber dynamics on trees and hyperbolic graphs. *Probability Theory and Related Fields*, 2005.

M. Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv*, 2017.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. In *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

N. de Freitas, P. Højen-Sørensen, M. I. Jordan, and S. Russell. Variational mcmc. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

P. Diaconis and L. Saloff-Coste. Comparison techniques for random walk on finite groups. *Annals of Probability*, 1993.

J. Ding, E. Lubetzky, and Y. Peres. Censored Glauber dynamics for the mean field Ising model. *Journal of Statistical Physics*, 2009.

V. Dinh, A. Bilge, C. Zhang, and F. A. M. IV. Probabilistic path hamiltonian monte carlo. In *International Conference on Machine Learning (ICML)*, 2017.

J. Djolonga and A. Krause. From MAP to marginals: Variational inference in bayesian submodular models. In *Neural Information Processing Systems*, 2014.

J. Djolonga, S. Jegelka, S. Tschiatschek, and A. Krause. Cooperative graphical models. In *Neural Information Processing Systems (NIPS)*, 2016a.

J. Djolonga, S. Tschiatschek, and A. Krause. Variational inference in mixed probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, 2016b.

S. Fujishige. *Submodular Functions and Optimization*. Elsevier Science, 2005.

B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Neural Information Processing Systems (NIPS)*, 2014.

A. Gotovos, H. S. Hassani, and A. Krause. Sampling from probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, 2015.

C. Guestrin, A. Krause, and A. P. Singh. Near-optimal sensor placements in gaussian processes. In *International Conference on Machine Learning (ICML)*, 2005.

T. Hazan, S. Maji, and T. Jaakkola. On sampling from the Gibbs distribution with random maximum a-posteriori perturbations. In *Neural Information Processing Systems (NIPS)*, 2013.

R. Iyer and J. Bilmes. The submodular Bregman and Lovász-Bregman divergences with applications. In *Neural Information Processing Systems (NIPS)*, 2012.

R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *International Conference on Machine Learning (ICML)*, 2013.

S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 1993.

M. Jerrum, J.-B. Son, P. Tetali, and E. Vigoda. Elementary bounds on Poincaré and log-Sobolev constants for decomposable Markov chains. *Annals of Applied Probability*, 2004.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 2012.

D. A. Levin, M. J. Luczak, and Y. Peres. Glauber dynamics for the mean-field Ising model: cut-off, critical power law, and metastability. *Probability Theory and Related Fields*, 2008a.

D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008b.

C. Li, S. Jegelka, and S. Sra. Fast mixing markov chains for strongly Rayleigh measures, DPPs, and constrained sampling. In *Neural Information Processing Systems (NIPS)*, 2016.

R. M. Neal. Mcmc using hamiltonian dynamics. *arXiv*, 2012.

A. Nishimura, D. Dunson, and J. Lu. Discontinuous hamiltonian monte carlo for models with discrete parameters and discontinuous likelihoods. *arXiv*, 2018.

A. Pakman and L. Paninski. Auxiliary-variable exact hamiltonian monte carlo samplers for binary distributions. In *Neural Information Processing Systems (NIPS)*, 2013.

G. Papandreou and A. L. Yuille. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *International Converence on Computer Vision (ICCV)*, 2011.

P. Rebeschini and A. Karbasi. Fast mixing for discrete point processes. In *Conference on Learning Theory*, 2015.

C. Sminchisescu and M. Welling. Generalized darting monte carlo. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

S. Tschiatschek, J. Djolonga, and A. Krause. Learning probabilistic submodular diversity models via noise contrastive estimation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Y. Zhang, C. Sutton, A. Storkey, and Z. Ghahramani. Continuous relaxations for discrete hamiltonian monte carlo. In *Neural Information Processing Systems (NIPS)*, 2012.

M. H. Zhu and S. Ermon. A hybrid approach for probabilistic inference using random projections. In *International Conference on Machine Learning (ICML)*, 2015.